

September/October 1988 Edition

RE_≡RUN

RUN Programs on Disk

For the C-64 and C-128

Recreate Famous Battles on Your C-64

RUN

COMMODORE 64/128 Users' GUIDE

U.S.A. \$2.95
CANADA \$3.95
U.K. £2.00

You Can Become a Power User!
How to Turn Your Fantasy Into Reality

Attention, New Owners:
Guidelines on Getting Started

Plus:

- Inertia Maria Arcade Action
- Card Shark 64/128
- Spelling Challenge
- C-64/128 Vacation Checklist

October 1988
Ad 10004
Publication

RUN

THE COMMODORE 64/128 Users' GUIDE

U.S.A. \$2.95
CANADA \$3.95
U.K. £2.00

Powerful Printers
Which One for You?

MUSIC EDITOR
Tunes Up
Your C-128!

MONKEY SEE
Fun for All Ages!

EDUCATION EXTRA:

- Software Reviews
- Instant Test Maker
- Teacher's Pet

Plus: Extra Bonus Program!

Introduction

September–October '88 ReRUN

THERE ARE 14 PROGRAMS in this September–October edition of ReRUN, with a good mix of C-64 and C-128 programs, plus some that operate in both 64 and 128 modes. I think you'll find their quality is top-notch.

In the first September program, Try To Remember, for the C-64, you're briefly shown a list of five words to commit to memory. The list quickly disappears, and one of the five words reappears as a string of dashes, with only one letter filled in. You must try to recall the word and, using your joystick, select the missing letters in their correct order from an alphabetical grid. I think you'll find it a challenging program for computerists of all ages.

Next, we're offering a great C-64 programming utility called Programmer's Pager. This interrupt-driven program allows you to display a Basic program one screen at a time, and it also features a highlighting bar to highlight lines of code.

Next, for the entertainment of card players, there is Knock!, a 64/128 variation of the popular card game, 31. Your opponents reside in the computer, and you'll find them a clever bunch of players indeed.

Inertia Mania, one of the largest purely machine language C-64 arcade programs ever published in a computer magazine, is also from our September issue. You're given a limited amount of time to guide a bouncing ball down corridors while collecting gold pieces and dodging cannon fire. This is arcade excitement at its best.

Our September Easy Applications program, Vacation List Generator, for either the 64 or the 128, creates exactly what its name implies: a checklist of camping and vacation gear. You can easily customize the list so it details your personal camping or vacationing belongings.

September's Mega-Magic program, Kracking the Kernal, is an exciting tutorial on programming. Veteran programmer John Ryan gives advice and examples on putting the C-64's powerful Kernal routines to work.

October's programs begin with Songs in the Key of C-128, a

fantastic music editor for your C-128 in 80-Column mode. You can compose, play, record and alter music with this flexible program.

Instant Test-Maker is a C-64/C-128 program that allows teachers to quickly create and print out tests and quizzes consisting of multiple choice, fill-in-the-blanks, true and false questions, and even essays. I know that many teachers will be delighted (and relieved) to get their hands on this wonderful time-saver.

Another October program is Easy Banners, a versatile banner-making utility for both the C-64 and C-128. It allows virtually any printer to create multiple-sized banners for promotions, parties and other events.

Monkey See, a C-128 concentration program, is played by first listening to musical notes and watching them appear, then trying to duplicate the sequence. On the same lighter side, there's Scared Silly!, a 64/128 pumpkin-producing program that should prove very popular this Halloween.

October's Easy Applications is Tool for Teachers, a z-scores calculator program for the statistical analysis of test scores. Along with Instant Test-Maker, it should put Commodore-equipped teachers in top form for this coming school year.

Our October Mega-Magic is Joystick Mouse Emulator, a C-64 program that demonstrates techniques for using your joystick as a mouse.

Finally, there's a bonus program, Flipper 34. This exciting program allows you to create and save up to 34 screens, or "pages," to disk. These can be loaded into memory and displayed, at a user-controlled speed, to create animation effects.

Well, that does it for this edition. Until next time, may your software and hardware problems be few.


Tim Walle
Technical Editor
RUN magazine

How To Load

LOADING FROM MENU

To get started, C-64 users should type LOAD "MENU 64",8 and press the return key. When you get the Ready prompt, the menu is loaded and you should type RUN to see a list of the programs on your disk. C-128 users need only press the shift and run-stop keys. When all the programs are displayed on the screen, you can run the one you select by pressing a single key.

LOADING FROM KEYBOARD

If you do not wish to use the menu program, follow these instructions.

C-64: To load a C-64 program written in Basic, type: LOAD "DISK FILE-NAME",8 and then press the return key. The drive will whirl while the screen prints LOADING and then READY, with a flashing cursor beneath. Type RUN and press the return key. The program will then start running. To load a C-64 program written in machine language (ML), type: LOAD "DISK FILENAME",8,1

C-128: All C-64 programs can be run on the C-128 as long as your computer is in C-64 mode. All C-128 programs are clearly labeled on the directory page. Your C-128 *must* be in C-128 mode to run these programs. To load a C-128 mode program, press the F2 key, type the disk filename and then press the return key. When the program has loaded, type RUN.

MAKING COPIES OF ReRUN FILES

Many programs on your ReRUN disk have routines that require a separate disk onto which the program writes or saves subfiles. To use these programs, you must first make a copy of the original program onto another disk that has enough free space on it to hold these newly written subfiles.

It's simple to make a copy of a Basic program. Just load it into your computer as outlined above, and then save the program back onto a separate disk that has plenty of free space for extra files.

Copying an ML program is not so simple. You cannot simply load and save an ML program; you'll need to use a disk-backup utility program, such as the one on your Commodore Test Demo disk.

Directory

PAGE	DOCUMENTATION	DISK FILENAME	FILE TYPE
		*MENU 128 _____	BASIC
		MENU 64 _____	BASIC
1	TRY TO REMEMBER	TRY TO REMEMBER _____	BASIC
2	PROGRAMMER'S PAGER	PAGE LISTER _____	BASIC
3	KNOCK!	CARDS 31 _____	BASIC
4	INERTIA MANIA	+ BANDIT BALL ML _____	ML
		BANDIT HI-SCORES _____	BASIC
		BANDIT BALL DATA _____	BASIC
6	VACATION LIST GENERATOR	VACATION LISTS _____	BASIC
7	KRACKING THE KERNAL	KERNAL PRG#1 _____	BASIC
		KERNAL PRG#2 _____	BASIC
		KERNAL PRG#3 _____	BASIC
8	SONGS IN THE KEY OF C-128	*MUSIC EDITOR 128 _____	BASIC
10	INSTANT TEST-MAKER	QUESTION BOX _____	BASIC
14	EASY BANNERS	EASY BANNERS _____	BASIC
15	MONKEY SEE	*MONKEY SEE _____	BASIC
16	SCARED SILLY!	SCARED SILLY _____	BASIC
17	TOOL FOR TEACHERS	Z-SCORES _____	BASIC
		Z-SCORES#2 _____	BASIC
18	JOYSTICK MOUSE EMULATOR	JOYSTICK POINTER _____	BASIC
		JOYSTICK#2 _____	BASIC
19	FLIPPER 34	£FLIPPER 34 _____	BASIC
		FLIPPER#2 _____	BASIC

* — C-128 mode only

£ — Bonus program

Before you run a program, carefully read the documentation that pertains to it.

Try To Remember

By Penny DeGroff

I REMEMBER IS A COMBINATION spelling and memory-improvement game that challenges you to remember and correctly spell lists of words in the shortest possible time. The nine difficulty levels each contain five words, with two-letter words at the first level, three-letter words at the second, and so forth. The game can be customized to fit anyone's ability.

When you run the program, it briefly displays the five words at the first difficulty level; then the playfield appears. This field is a matrix of all the letters of the alphabet, with a light-blue ball in the center. Above the playfield is the time display. The computer picks one of the five words, then shows one clue letter from that word below the playfield, with the other letters represented by hyphens.

The clock is ticking, so you'd better start. Using a joystick in port 2, move the blue ball to the letter you think the first hyphen represents. Take care to move the ball in the spaces *between* the letters, for the ball turns yellow as soon as it touches *any* letter, and when it does, the die is cast: you have chosen *that* letter.

You can proceed to another letter only by guiding the (now) yellow ball to the star in the center of the playfield, at which point you hear a happy ping if your choice is correct, a dull bong if you're wrong. If you've chosen correctly, the letter will replace the hyphen; if not, try again.

Continue until the entire word is displayed. Then you'll advance to the next level.

CUSTOMIZING THE PROGRAM

To change the length of time that the lists of words are shown, increase or decrease the 4000 in line 260 of the program. If you want to play only with shorter words, decrease the 10 in line 240. For example, if you'd like the game to end after the six-letter words, change the 10 to 6.

You may want to replace some or all of the words from time to

time. They're located in Data statements beginning at line 630, with the two-letter words in line 630, the three-letter words in line 640, and so forth. When you replace words, be sure they're the correct length for that level and that you keep five in each Data statement. I'd also recommend that, within each level, no two words have the same letter in the same position. For example, if "but" and "pet" were in a list, the program might select one of them and "t" as the clue letter. Then, you couldn't guess the word until you'd chosen more letters, possibly wasting time.

RUN it right: C-64

Programmer's Pager

By John Ryan

PAGE LISTER IS A POWERFUL, albeit short, programming utility that takes the hassle out of listing Basic programs to the screen for study and debugging. No more will you have to type in LIST <line number> - <line number> over and over again to search for bugs. At the touch of a function key, Page Lister brings 24 lines of Basic text to the screen and leaves them there for your perusal and editing. As a bonus, a line-highlighting function is available to help eliminate the tedium of scanning for errors. Anyone who's tried to compare data between a magazine listing and the screen or who works with Data statements a lot will love this feature!

When you want to use Page Lister, load and run it to generate the machine language program in its Data statements; then load the program you plan to work on. With both programs in memory, just tap the restore key to activate the Page Lister features. If you've forgotten to load your other program, a message appears to remind you. Otherwise, the screen goes blank and awaits your next keypress.

Press F1 each time you want to examine a new batch of 24 lines. At the end of the listing, the next F1 will begin recycling at the first line number. To page backward through the listing, press F3. In this case, when it reaches the beginning of the listing, the paging stops; there's no cycling with F3.

F5 activates the highlight bar, the cursor keys move it up and down and another F5 toggles it off. Pressing F7 exits from Page Lister.

This utility works fine with all Basic programs except those with embedded list-protection features and those with machine language programs automatically appended to them by certain assemblers.

RUN it right: C-64; C-128

Knock!

By John Fedor

"TLL BET YOU A NICKEL!" Okay, but how about \$5? The winnings are up in Cards-31, a Commodore version of the traditional game in which you try to win a "pot" of nickels by getting a total score of 31, or as close to it as possible, with cards of the same suit. Numbered cards are worth their face value, face cards are worth 10, and aces count 11. In this game, a pot of imaginary \$5 bills is at stake instead of nickels.

You can play up to four computerized opponents in Cards-31, or just sit out and watch them battle among themselves. The game starts with each player having three cards and \$15. A discard pile is established with one card facing up. When it's your turn, you have four options: draw a card from the deck; draw the exposed card from the discard pile; knock (which you do when you're as close to 31 as you figure you can get); or ask your expert opponent (the computer) to look at your cards and offer advice. If you ask for advice, don't worry—the machine won't remember your hand when playing its own.

If you draw from either the deck or the discard pile, you must discard to keep only three cards in your hand. In general, drop a card that's not of the same suit as two others in your hand or that's of less value than the others. For instance, if you hold a jack and queen of spades and a king of diamonds and then draw a 9 of spades, drop the king of diamonds. If you hold a 7, jack and king of hearts and draw a 10 of hearts, drop the 7.

You can choose to knock when you feel you've reached the best score you're going to get, or when you chance to get a fair score early in the round and judge that your opponents may not yet have collected much in one suit. After someone knocks, each of the other players gets one more turn to better his hand. This ends the round, and the scores are computed and displayed with the final hands.

Only the cards in the suit each player holds the most of count in the scores of each round. For example, if you hold a 9 and 4 of hearts and a 10 of clubs, your score is 13 (9 + 4). If your cards are all in different suits, only the card with the highest value is counted. The maximum you can get for your three cards is the magic 31 (the ace plus two face cards or one face card and the 10, all in the same suit).

As a round ends, each player whose score is less than the best loses \$5 to the pot, and the amount in the pot is displayed, as well as the amount each player has remaining. The next round starts with the player after the one who knocked last. When you're out of money, you're out of the game, which continues until only one player has any money left—and he wins the pot!

You'll find your computerized opponents the most skillful players this side of Las Vegas, and each has different instructions for when to knock. With the variety of play options available and the strategy involved, this version of 31 should keep you entertained for hours.

RUN it right: C-64; joystick

Inertia Mania

By Charles Orcutt

IF YOU'VE NEVER BEEN in a rubber room, Rubber Bandit Ball provides a golden opportunity to feel what it's like. In this smooth-scrolling, multiscreen arcade game, you use a joystick in port 2 to control a rubber ball as it bounces through corridors, trying to snatch up little golden nuggets. Guiding the ball is hard enough, but merciless cannons fire killer balls through the corridors periodically, and the rubber ball loses one of its five lives if it collides with one. A clock allows you 99 seconds to gather up all 95 nuggets, and

the ball also loses a life if it runs out of time. When the ball does get all the nuggets, you proceed to the next level, where the cannons fire faster killer balls more frequently.

For each nugget snatched, you get the number of the current level in points. In other words, in level two, each nugget is worth two points. You pause the game by hitting the fire-button, and you start over again by pressing restore.

Like most objects in this universe, the rubber ball has inertia, so it tends to continue in the direction it's going. As a result, moving the joystick is like blowing on the ball or pulling it with a spring; the effect is subtle and difficult to predict. Sometimes the ball gets mired in the wall (especially near the cannons), so be careful. You'll eventually learn how to avoid getting stuck.

The high-score display on the game screen is valid only for the current session of play. Your top-ten high scores are stored, in descending order, on disk. When you run out of lives and play ends, the game boots the high-score file and checks to see whether your current score should be included. If so, enter your initials by moving the joystick up and down to cycle through the alphabet and left and right to move between your initials. When you press the fire-button, the new top-ten roster is saved and another game commences. The high-score routines of some games have you press the fire-button as you enter the letters. Be sure not to do that here, or you won't get your initials placed.

In order to save Rubber Bandit Ball scores to disk, you must copy Rubber Bandit Ball and the program called Hi-Score to a work disk. Otherwise, the game won't save the top-ten list, and you'll get garbage on the screen.

Don't use a fast-load cartridge with Rubber Bandit Ball. My cartridge causes errors when the program tries to store the high-score file, and I couldn't figure a way around the problem. Just use the standard Commodore DOS.

RUN it right: C-64; C-128 (40- or 80-Column mode); printer

Vacation Checklist Generator

Program by Lou Wallace; Article by Tim Walsh

HOW DO YOU SPEND your well-deserved vacations? Whether you bask in the tropical sun of Antigua, stalk wild boar in the Maine Alagash or just pitch a couple of tents in the local swamp for a Saturday night of mosquito-slapping, you could use Vacation Checklist Generator.

As its name implies, Vacation Checklist Generator allows you to produce a list of the items you're going to take on your trip. Many camping items are included on the list and automatically appear when the program is run. You simply need to answer Y or N at the prompts to add or skip an item on your list of equipment. Once it's completed to your satisfaction, you can print it out.

When packing your car or suitcases, you check off each item against your printed list. Not only are these lists invaluable for packing when you leave home, but also for checking to see you haven't forgotten anything when you're leaving your campsite. This reduces the risk of not-so-funny whines on the ride home, such as, "Dad, I think we left the tackle box behind." You can reassure yourself and your offspring by showing that you checked off the tackle box beforehand.

As written, the program's camping gear list was formed with the no-holds-barred camper in mind. Few vacationers have a need for a shotgun, hunting knife and camouflage when traveling to Disneyland, so the program permits you to skip any or all of the default camping equipment and create your own customized list by substituting items.

As a notoriously ill-prepared camper, I view this program as the ultimate camping accessory. Never again will you accidentally leave anything behind, except, hopefully, your worries.

Kracking the Kernal

By John Ryan

MANY BEGINNING PROGRAMMERS assume the C-64 Kernal is filled with 6502 subroutines that are as mysterious as hieroglyphics and used exclusively by machine language programmers. While this may be true to some extent, anyone can use Kernal routines to access the power and speed of the built-in ML routines without having to learn machine language.

The Kernal Plot routine lets you position the cursor anywhere on the screen without using a lot of cursor commands. The three routines below show how to use the Plot routine in conjunction with the Kernal's line-clearing, line-insertion and screen-scrolling routines.

Kernal 1 utilizes both the Kernal Plot routine and the Kernal routine that clears a screen line. This is useful for programs that need to clear only a portion of the video screen. It also clears the screen from the bottom up. To clear the screen from the top down, change line 30 to `X=0:Y=0`; then reverse the logic in line 60. If you want to use just the Plot routine at 65520 (\$FFF0), set `X=row (0-24)` and `Y=column (0-39)`; then place line 100 as a subroutine in your own program.

Kernal 2 shows how to achieve rapid screen scrolling from Basic by using the Kernal routine at 59626 (\$E8EA). This program prints a line of text at the bottom of the screen, then scrolls it to the top.

Finally, Kernal 3 scrolls text down the video screen. The Kernal routine at 59749 (\$E965) is a subroutine the Kernal uses to insert a blank line on the screen. You can see this by listing a program, going to a line with fewer than 40 characters, then adding text until the 40-column boundary is crossed. Notice the extra blank line that appears.

Program 3 is called with the invisible cursor positioned at the top of the screen, just above the printed text. When the Insert routine is called, the computer automatically moves everything on the screen down as it adds a blank line.

The Kernal is for everyone who wants to squeeze every ounce of power from his C-64 without having to re-invent the wheel. Grab a memory map and explore the Kernal for yourself. Who knows what little ML gems you'll find for your Basic programs!

RUN it right: C-128, in 80-Column mode; joystick; printer optional

Songs in the Key of C-128

By Barbara Schulak

BETTER TUNE UP YOUR C-128! Music Editor 128 facilitates transcribing written music, whether your own or sheet music, into strings that can be used by the C-128 Play statement. It will also store the music as a sequential file on disk, so you can rework it later or use it in your own programs.

The Music Editor screen displays three menus and octaves 1-6 of the piano keyboard. The two bottom lines of the screen are reserved for input and program response.

To select options from the menus, use a joystick plugged into port 2. Move it up and down to highlight the choices; then press the fire-button to select the one you want.

Menu 1 lets you access the editor (which leads to menus 2 and 3), load or save a file, print the file either to the screen or your printer, view the directory and set the tempo for the music. After you press the fire-button, a prompt appears at the bottom of the screen. Enter the necessary information, and the operation will proceed. To abort the operation, press the return key. You can also play all or part of the music from menu 1 by entering the starting and ending measures you want to hear.

KEYING IN THE MUSIC

To facilitate entering data, Music Editor works one measure at a time. Menu 2 lets you choose a measure to edit, start and end the measure and play it. You can also select the volume for each measure and the tuning envelope for each voice. I'd suggest using the first measure to define your tuning envelopes, so they can easily be

changed later. It's also possible to change the tuning envelopes and volume for sections of the music. When you are all ready to begin work, choose the Start Measure option, which takes you to menu 3.

Menu 3 is for entering notes. First use the joystick to highlight and select the Voice Number option. Then move the joystick up and down and press the fire-button to choose the voice number, 1-3. Notice that the number you've chosen appears at the bottom of the screen. This is where the string will be displayed as you enter it.

To enter a note, move the joystick up and down to choose the duration (whole, half, quarter note, and so forth), then right and left to slide the arrow up and down the keyboard. When the arrow is pointing to the note you want, press the fire-button. The note will be played and displayed at the bottom of the screen. The red R on the keyboard stands for a rest. Continue in this fashion until you've completed your measure.

Then return to menu 2 to play the measure. If it sounds fine, select End Measure to add it to your file. If not, you can begin again by selecting Start Measure, which erases the string, or return to menu 3 to edit the string. Place the arrow on the red E at the right end of the keyboard and press the fire-button to erase a letter from the end of the string. You may have to do this several times to erase the parts you want to delete. When your measure is finished, select End Measure from Menu 2.

You can choose any measure to edit by selecting the Measure # option. Be sure to save your file periodically as you work.

PROGRAM MUSIC

When you have a piece finished and want to include it in a program, use the following code to load and play the file:

```
10 F$ = "FILENAME":DIMAS$(200)
20 DOPEN#2,(F$),R
30 INPUT#2,N
40 FORI=1TON:INPUT#2,A$(I):NEXT
50 DCLOSE#2
60 FORI=1TON:PLAYA$(I):NEXT
70 END
```

You can add your own Tempo statement (the default is 8), change the tuning envelopes with the Envelope statement and add filtering effects with the Filter statement. The C-128 reference guide's discussion of music commands will help you use Music Editor.

Keep in mind that Music Editor limits strings to 160 characters (two screen lines). You may have to divide a measure into two sections if your string is getting too long.

RUN it right: C-64; C-128 in 40-Column mode; printer

Instant Test-Maker

By Hugh McMenamin

OF ALL THE CHORES teachers have to face, the most time-consuming seems to be preparing and grading examinations. The Question Box program eliminates some of this drudgery by letting you construct a library of examination questions and then retrieve individual questions to assemble into tests.

Question Box, which works on both the C-64 and the C-128 in 40-Column mode, handles all the common forms of questions: multiple-choice, true-false, fill-in-the-blank and essay. Individual questions can be previewed from the Master file and selected either by you or, randomly, by the computer. The new Test file can then be printed out or stored to disk. As a bonus, you can include superscripts and subscripts in your questions.

Here are instructions for using the Question Box's numerous commands.

ENTERING QUESTIONS

E (Enter) calls up the menu for selecting the type of question. The body of a question may consist of four lines of up to 60 characters each, with a — marker in reverse field indicating the 60-character limit. If you exceed the limit, the program will truncate the line automatically.

To enter each line, press the return key. Also press return to enter null (empty) lines when no further text is desired. If you want additional space between the body of a multiple-choice, true-false or essay question and the answer, enter a shift/space.

Note that commas and colons *may* be used in your entries, but quotation marks may not.

T (True-false) questions are automatically followed with the two responses to be circled: True False

F (Fill-blank) requires you to construct the blank, (____), within the question by using the Commodore and @ keys simultaneously.

E (Essay) mode provides for up to nine empty lines after the main body of the question.

M (Multiple-choice) questions allow you to add four answers after the letters A-D, again within the 60-character limit per line.

Each time you finish entering a question, you'll be asked if you want to add another. If your response is other than Y, the command screen will return.

R (Review File) lists the number of each question in the Master file and 26 characters of the question's first line. The question type is also shown, in reverse field, for easy identification in using the Correct, Delete and Transfer modes.

C (Correct) asks for the number of the question to be corrected, then displays the question, along with code numbers 0-7 for multiple-choice and 0-3 for the others. Null lines are indicated by a character space and period after the number (5 .), while shift/space lines are indicated by a blank after the number (5). The line you select by entering a code number is displayed below a dashed line. Make your corrections, including commas and colons, then reenter the line by pressing return.

MAKING UP TESTS

T (Transfer) mode lets you select individual questions to be transferred from the Master file to the Test file. If you don't enter N at the prompt, the selected question will be appended to the Test file.

N (New Test) moves a random selection of questions from the Master file to the Test file. Both Transfer and New Test limit Test file questions to 50. If New Test doesn't reach the 50-question limit, you can add other questions to the Test file individually.

P (Print) allows printing from either the Master or Test file to the screen, disk or printer. When printing to the screen, the display will scroll about 20 lines. When printing to the printer, only complete questions will appear on a page; they won't wrap to the next page. Print-to-disk stores the data as ASCII characters, which then can be loaded into a word processor, such as RUN Script, for further revision. If you're using RUN Script, you can enter formatting codes with the Define Macros program (see *RUN*, January 1987). The print-to-disk files are identified by a hyphen (-) preceding the filename.

D (Delete) asks for the number of a question to be removed from the Master file. Since the remaining questions are renumbered, check first with Review for the correct numbers of any additional questions to be deleted. Once a question has been deleted, it cannot be recalled.

DISK OPERATIONS

Press **4** or **\$** (shifted 4) to view a directory of all the Question Box files on a disk. These files are identified by a **+** before the name.

L (Load) displays a **\$** prompt to ask for the name of the file. Press return to view a list of the Question Box files on the disk, then select the one you want by moving the cursor to the name and pressing return. Or, you can type in the filename without the **+** prefix.

You'll also be asked if the file should be added to the Master file. With this mechanism, you can append small files to each other to build larger files. Just don't try to make the Master file too big; the Question Box limits it to 100 questions.

S (Save) asks if the Master file or the Test file is to be saved. When a Test file is saved, it is appended to the filename as an identifier. You can load Test files directly from the disk to the Master file by entering the entire name, including the **.t** suffix.

QUIT AND HELP

Q (Quit) terminates the program. If you press **Q** accidentally, you can preserve any test data in memory by entering **GOTO 190** and return.

H (Help) lists the Question Box commands and some of the program functions.

SUPERSCRIPTS AND SUBSCRIPTS

I designed the Question Box so that you can utilize embedded super- and subscript commands with printers that support them. You have two size options for these: conventional and full-size. Those of the conventional small size are set by line 160. The control code for the superscript is **CHR\$(27) + "S0"**, and for the subscript **CHR\$(27) + "S1"**. Both are cancelled by **CHR\$(27) + "T"**.

To activate superscript printing, press the up-arrow (**↑**) key. After the superscript is completed, press the left-arrow (**←**) key to return to normal printing. The subscript is activated by the British pound key (**£**), and again is cancelled by the left-arrow key. Following this pattern, the formula for water would be printed with **H£2-O** and the notation for ten squared with **10↑2←**.

Because of the difficulty in copying the miniature characters with some printers, I've provided for full-size super- and subscripts. However, they work only with printers that allow you to reverse direction with the paper feed. Examples of these are the Epson FX-80 and emulators of the FX-80 series (the C.Itoh Prowriter Jr. and the Star NX-1000, in Tractor mode only).

Full-size is selected at the beginning of the printer option. The printer control codes are defined as the up string (up\$) and down string (dn\$) in line 140, and these strings are activated when you choose the full-size option.

To initiate full-size superscripts, you press the \uparrow key, and you cancel them with the \pounds key. Subscripts are activated and cancelled with the reverse procedure: \pounds to activate, and \uparrow to cancel. With the full-size characters, then, you'd print ten squared with $10\uparrow 2\pounds$ and the formula for water with $H\pounds 2\uparrow O$.

OPERATING NOTES

The Question Box is heavily dependent on string manipulation, which slows operation on the C-64 because of that machine's memory limitations. You can speed things up by placing a REM at the beginning of line 240 to negate the Fre(1) command, which performs the nonessential functions of calculating free space left and packing strings.

If your computer seems to stop while you're running the Question Box, it's probably the result of garbage collection and not a complete lock-up. When this happens, wait for a minute or two to see if the program continues. If not, press run-stop/restore and enter GOTO 190 to preserve your questions.

If you possess Abacus's Basic 64, I'd highly recommend using it to compile the Question Box. Then it will work very well in C-64 mode, even with line 240 fully operational.

If your printer won't work with the Question Box, you may have to change the control codes in lines 140-170 to accommodate your interface. See your printer manual for these codes.

Last but not least, be sure to save your questions to disk frequently. Then you won't have any test surprises!

Easy Banners

By John Ryan

WHETHER IT BE A BIRTHDAY PARTY for your five-year-old or a welcome-home celebration for a loved one, almost everyone needs to create a quick sign or banner at one point or another. While most printer utilities limit you to printing messages on a single sheet of paper, Easy Banner will print any message sideways, enabling you to create banners as long as your imagination or pocketbook allows.

Although the program is deceptively short, Easy Banner is flexible enough to meet the needs of most occasions. It may be loaded in either C-128 40-Column mode or in C-64 mode, and there are four options in the program. The first lets you choose the height of the characters you wish to print (1-8, smallest to largest). Then, under this option, you may wish to elect an expanded print character at the prompt, since the largest sizes (7 and 8) tend to be vertically elongated and thus hard to read at close range.

The second option lets you specify the character you wish to print your banner with; this may be any Commodore keyboard character. Finally, select option three to enter the message you want printed, first making sure that the printer is connected and turned on, and that any special features have been disabled (most notably, skip-perforation DIP switches). The fourth option lets you exit the program.

Easy Banner will handle inputs of up to 80 characters at a time, which, believe me, will use a tremendous amount of paper. Of course, you may create longer messages by simply appending separate banners.

The program should be compatible with most dot matrix and letter-quality printers. So next time, don't paint it. .print it!—with Easy Banner.

Monkey See

By Jerome Reuter

YOU'RE PROBABLY FAMILIAR with the game called Simon Says, or Follow Me. You may even have one of those little plastic boxes with four buttons on it—and a dead battery inside—lurking in a closet. Well, here's a program for the C-128 that brings the game to life. Called Monkey See, it replaces the plastic box with the 40-column screen and the C-128's keypad. Instead of a blinking colored light, you see animated sprites on a colorful background, and the sound is much superior to the buzz of the original game.

When you run the program, it boots up with the difficulty level set at E, for easy, but you can change that to medium, hard or impossible by pressing any key to cycle through the options, and then the return key to select your challenge.

The screen shows four boxes, or "pads," arranged like the 2, 4, 6 and 8 keys on the keypad, with a musical note sprite in the area in the middle. The computer plays a note by "pressing" on one of the pads with the movable note sprite. Then it waits for you to play the same note with keypad number 2, 4, 6 or 8. After you play the note, Monkey See ups the ante by playing the first note again, and then another. This pattern of adding notes continues until you make a mistake or exhaust the level you've preset.

If you make a mistake, the screen offers the options of hearing the last sequence played correctly, starting the game again or quitting.

Monkey See is written in Basic, with Data statements at the end for the sprite data and the actual key sequence. Don't think for a moment that the game won't be random enough with the key sequences in an array. While the array holds a sequence of only 100 keys, there are over 50 random starting positions.

The listing lines are numbered by two's, so you can use your auto numbering system and still have room to add custom comments where needed. The difficulty levels are set in lines 98-104 and should provide the amount of challenge you need. The sprite movement is set in lines 88-96. A variable empty timing loop in those lines

keeps shortening the time between notes until either you or the computer give up.

One caution: Monkey See is good fun for all ages, so don't turn up the volume too loud. Otherwise, you'll have a number of people standing behind you, all yelling, "I'm next!"

RUN it right: C-64; C-128 (in 40-Column mode)

Scared Silly!

By Joey Latimer

ONE CHARACTER YOU WOULD not want to be on Halloween is a pumpkin. You'd first suffer the horrible ordeal of getting your insides scooped out and your eyes, nose and mouth carved. Then you'd be placed outdoors, helpless in the black night and chill wind and at the mercy of ghosts and goblins and other terrifying monsters who want to squash you.

So, instead of making a scary pumpkin, I made a *scared* pumpkin!

His eyes dart from side to side in frantic desperation, hoping against hope that *this* house will be overlooked by the trick or treaters.

Without warning, a sickly white flash lights up the night, followed by the sharp slap of thunder. And music! Where's that haunting music coming from! Another bright flash silhouettes a trick or treater gliding up the walk. The porch light pops on. No! It can't be! It's wearing an Ozzy Osborne outfit! Arrrggghhh!!!

You can run Halloween Pumpkin on a C-64 or a C-128 in 40-Column mode. If you use C-128 mode, you'll notice that the pumpkin looks quite different.

Put your scared pumpkin in a window—safe from the wicked night outside—and hope the tricksters and goblins will be kind on this Halloween.

RUN it right: C-64; printer recommended

Tool for Teachers

By Rick Kephart

TEACHERS WHO ARE COMMODORE OWNERS will appreciate this Z-Scores program. It statistically analyzes test scores to produce a z-score for each student and then a more easily used "curved" grade derived from the z-score. It also generates a rank-ordered listing of the raw scores (the scores entered).

Z-scores are based on the standard deviation from the mean of the raw scores. Therefore, they normally remain quite constant from test to test, regardless of the difficulty of the test. When a student's z-scores are consistently lower than those of most other students taking the same tests, he or she probably needs special help. Similarly, if a particular student's z-scores suddenly drop, he or she may be encountering a momentary difficulty that requires attention.

Z-scores can indicate problems with a test, as well. If many of the z-scores for the test are inconsistent with the students' past records, the test design may have been faulty, or the students may have been guessing too much, or even cheating.

A z-score of 0 is average. Positive z-scores are above average, and negative scores are below average. To make the z-scores more meaningful, the program plots them on a standard bell curve, then translates them into letter grades. The letter grades have the following meanings: A, far above average; B, above average; C, average; D, below average; E, far below average. The program also displays the average of the raw scores.

You have a choice of an on-screen display or a printout. I recommend a printout, especially when you have more than ten students taking a test. Both outputs consist of four columns—name, rank ordering of the raw scores, curved letter grades and z-scores—with the average of the raw scores below.

Lines are skipped in the display or printout, except when identical raw scores have been earned. This makes the statistical mode—the most frequent score—in the rank ordering more apparent. If there's more than one mode—that is, if more than one score appears at

this greatest frequency—there may be a need for separate grouping.

In addition to your choice of screen display or printout, you have the option of using a sequential disk file for inputting students' names. This file is created by Z-SCORES PRG2. Having the names on disk saves retyping them each time you run Z-Scores. However, you can't use file input if you don't have a score for every name in the file.

When entering names and scores into the program, you can correct mistakes by typing - 1 at the next prompt. Pressing the return key without entering a name indicates that you're done with the names; pressing return while entering scores repeats a score.

Note that only the first 15 letters of a name will appear in the screen display or printout. Also, scores must represent the number or percentage correct, not those wrong; no score can be less than zero or more than 127; and only whole numbers are allowed.

RUN it right: C-64; joystick

Joystick Mouse Emulator

By Kenny Lawson

WITH A JOYSTICK IN PORT 2, Joystick Mouse Emulator moves a pointer about the screen much as does a mouse input device. Load and run EMULATOR 1, then load and run EMULATOR 2, which lists some typical commands on the screen, along with the pointer, calls the machine language routine and waits for the joystick to move the pointer. A formula computes the pointer's screen location (1024-2023 in default) when you press the fire-button.

The pointer moves in the direction you press the joystick. When you move the tip of the pointer over a desired instruction, press the fire-button to get a message identifying your selection. For instance, if you move the pointer to the Print option and press the fire-button, the screen displays the message "You called for Print".

The data for the sprite shape begins at the eighth data element in EMULATOR 1 on line 220. Change this data to create other shapes. Data items 13 and 15 in line 200 change the speed at which

the pointer moves; I used values that make it move at a reasonable speed.

You can use Joystick Pointer as a machine language subroutine in your own applications programs. Just use a Gosub command early in your own program to install the ML. Your programs will have to contain a series of If-Then statements, such as the demo program uses, to determine the action taken when you make a selection.

Of course, in using the joystick routine in your own applications, you must map out the format of your own on-screen instructions, commands, options, and so on. The formula in line 70 of the demo program returns the number of the screen cell that the pointer occupied when you pressed the fire-button, by reading the sprite horizontal and vertical registers for the pixel number of the line and column of the sprite. For instance, for a pointer in the upper-left corner of the screen, $P = 1024$, the first screen location in default memory.

To switch screen locations, adjust the formula by changing 1024 to the number of the first byte of screen memory that your program uses; the joystick routine still works.

RUN it right: C-64

Flipper 34

By Francis O. Saffell

FLIPPER 34 EXTRACTS the full potential of the little-used graphics characters of the C-64, and, like any good utility, it may be used in a variety of ways. As a stand-alone program, it creates and rapidly displays up to 34 pages of screen data that can be used to create short cartoons, op art or dazzling displays for window advertising.

As a programming tool for your own programs, Flipper will create page sets that you may recall and display by incorporating the Flipper subprogram.

HOW TO USE THE PROGRAMS

Flipper uses an editing screen for creating pages, which are re-

trieved (fetched) or sent (stashed) to and from this editing screen. The screen works like the normal Basic screen editor, with a few exceptions. The delete key now works as a backspace, and the insert key changes the characteristics of the cursor. Normally, Flipper 34's red cursor advances one space after placing a character. Pressing the insert key changes the cursor color to cyan and turns off the auto-advance feature, thus making it easier to create vertical lines. Pressing the insert key again returns the red cursor.

When creating pages with Flipper, you can use keyboard graphics characters. As most Commodore computerists know, holding down the Commodore key while pressing another key produces the graphics character depicted on the lower-left corner of the key. Holding down the shift key and pressing a key produces the lower-right graphics character. To repeat a character with Flipper, first place it on the screen, then bring the cursor over it and press shift/return. The character is now captured. To place it onto the screen, press the return key.

To simplify animation, newly placed characters are displayed in green. After you stash or fetch a page, all characters become white. Newly placed characters will then be green, which simplifies setting up new pages. Old white characters are quickly removed by pressing the control and 2 keys simultaneously.

While you're editing, the top line displays a page-in value, a page-out value and the range. The page-in value denotes the most recently fetched page. You use the function keys to control page movement. Press the F1 key to move a page into the editing space. When starting out, you'll be retrieving whatever was left in memory. You can clear the edit screen with the clear key.

You stash a screen by using the F3 and F5 keys. The page-out value is automatically incremented each time you press the F3 key, and this can be overwritten. The F5 key stashes the edit screen to the page-out location. The high-range value is automatically set to the highest stashed page. You can independently set the range values with the F2 key. The F4 key turns on the page-number and screen-address display feature. Here the otherwise inaccessible top line of the edit screen works to your advantage, because it automatically hides the top line during page-flipping.

Pressing F6 loads a previously saved set of pages and automatically sets the range values. The F8 key saves all pages within the start and finish range values as individually numbered program files. Be careful when saving. All files with names that start with letters

matching the page-set name will be deleted before the save begins. Pressing the F7 key starts the flipping routine. While pages flip, you can adjust the speed with the plus and minus keys. To return to the edit screen, hold down the F7 key until the top information line reappears, then quickly release the key. You press the £ key to see a display of the function key definitions. Pressing the £ key again returns you to the edit screen.

CREATING A TWO-PAGE ANIMATION

Load and run the Flipper program. Press the clear key and answer "y" to the prompt. Now press control/9. Push the Commodore/+ key combination, and then press control/0. You should then have a grid character in the upper-left corner. Now place the cursor over this character and press shift/return. Hold down the return key to fill the top line with this character. At the end of the line, press shift/insert to get the cyan cursor.

Now, using the cursor-down key and the return key, fill the right horizontal border with the grid character. At the bottom of the screen, press the home key. Now fill the left border with the grid character. When the cursor reaches the bottom, press the insert key again to get the red cursor. Hold down the return key to fill the bottom line with grid characters.

Now press the F5 key to send the contents of the screen to page 1. Press the home key to return the cursor to the upper-left corner; then press the Commodore/+ key combination. This places the normal, non-reversed grid character on the screen. Place the cursor over this character and capture it with the shift/return combination. Now place that grid character over the old, reversed characters with the return key.

When you are through, press the F3 key, followed by the return key, then finally the F5 key. This sends the positive grid characters to page 2. Press the F4 key and enter "y" for yes. This will display the page number and the screen address during animation. Now press the F7 key to view your two-page animation. You may alter the rate at which the pages change by holding down the plus or minus keys. To return to the screen editor, hold down the F7 key until the top display line reappears.

You can now save these two pages and display them using the Flipper subprogram. Press the F8 key and enter a unique five-character filename. After the save, press the stop key, clear the screen, enter NEW and load the subprogram. Run the subprogram and

enter the page set name. The subprogram loads and displays your animation until you hold down the F7 key to halt the routine.

MERGING THE SUBPROGRAM

The subprogram is easily merged with your own programs. Simply load and list it to the screen. Bring the cursor up under the last line; then type NEW and press the return key. Bring the cursor up under the last line again and load your own program. After your program loads, bring the cursor up to the top line and press return over each of the line numbers. The subprogram is now appended to the end of your program.

LET'S SEE A DEMO!

An animated demonstration of Flipper 34 is on this ReRUN disk for your viewing enjoyment. Just activate Flipper 34, press the F6 key and enter the filename "WOW!" at the prompt. Wait about three minutes for all 34 pages to load in. Once the loading message disappears, press F7 to begin the animation process. As explained earlier, press + to speed up and - to slow down the animation. Hold down the F7 key to halt the animation process.

PROGRAMMER'S NOTES ON PAGE SET FILES

Page set files are of two types: individual screens stored as program files, and sequential files containing, among other things, the filenames of all the screens in that set. The sequential ".dir" file has the start and finish range values of a page set. Individual page records comprise four fields. Field 1 contains a page program filename; field 2 contains the value for bank selection; field 3 contains the VIC II screen offset value; and field 4 contains the high byte of the absolute screen location in memory.

To maximize programming space at the start of your program, poke locations 52 and 56 with the high byte of the starting address of your lowest page; then issue a Clear command. If your pages start at 22528 to 31744, you'll also have to protect the character data at 20480 to 22527 by poking 52 and 56 with a value of 80. Location 648 can also use the screen-location high byte to direct printing onto your animation. To get a screen's location, multiply the screen-location high-byte value by 256. ■

RE = RUN

EDITOR IN CHIEF

DENNIS BRISSON

TECHNICAL MANAGER

LOU WALLACE

TECHNICAL EDITOR

TIM WALSH

MANAGING EDITOR

SWAIN PRATT

SENIOR EDITOR

BETH S. JALA

ASSOCIATE EDITOR

HAROLD R. BJORNSEN

COPY EDITOR

PEG LEPAGE

ART DIRECTOR

HOWARD G. HAPP

DESIGN AND LAYOUT

ANNE DILLON

TYPESETTING

DEBRA DAVIES

KEN SUTCLIFFE

FULFILLMENT CONSULTANT

DEBBIE BOURGAULT

14 Programs Included on this Disk

Try To Remember ▶ Programmer's Pager
Knock! ▶ Inertia Mania ▶ Kracking the Kernal
Instant Test-Maker ▶ Easy Banners
Tool for Teachers ▶ Joystick Mouse Emulator

From the September RUN:

- ▶ Try To Remember
- ▶ Programmer's Pager
- ▶ Knock!
- ▶ Inertia Mania
- ▶ Vacation List Generator
- ▶ Kracking the Kernal

From the October RUN:

- ▶ Songs in the Key of C-128
- ▶ Instant Test-Maker
- ▶ Easy Banners
- ▶ Monkey See
- ▶ Scared Silly!
- ▶ Tool for Teachers
- ▶ Joystick Mouse Emulator

Plus: Extra Bonus Program!

- ▶ Flipper 34

If any manufacturing defect becomes apparent, the defective disk will be replaced free of charge if returned by prepaid mail within 30 days of purchase. Send it, with a letter specifying the defect, to:

ReRUN • 80 Elm Street • Peterborough, NH 03458

Replacements will not be made if the disk has been altered, repaired or misused through negligence, or if it shows signs of excessive wear or is damaged by equipment.

The programs in ReRUN are taken directly from listings prepared to accompany articles in *RUN* magazine. They will not run under all system configurations. Use the RUN It Right information included with each article as your guide.

The entire contents are copyrighted 1988 by IDG Communications/Peterborough. Unauthorized duplication is a violation of applicable laws.

©Copyright 1988 IDG Communications/Peterborough

